Improving the Performance of Adopted Approaches for Extracting Arabic Keyphrases

Fatma Elghannam
Electronics Research Institute,
Cairo, Egypt

ABSTRACT
In this work the improvement of automatic keyphrases extraction using deep linguistic features and supervised machine learning algorithm is discussed. The n-gram method for extracting important keyphrases produces huge number of candidate terms. Many of those terms are non-keyphrases either because they are linguistically non expressive terms or due to redundancy in sense. The objective is to restrict the number of candidate terms and keeping the relevant ones. This work is an extension to a previous one in keyphrase extraction for Arabic documents. The proposed work covers the deep linguistic features of the candidate terms. To capture the well-structured terms a new-added definite structure feature is introduced and tested. A set of linguistic features of the previously assigned candidate terms are applied to a supervised machine learning technique to classify the candidates as keyphrases or not. The experiments carried out showed that the proposed technique improves the accuracy of extracting keyphrases relative to the previous version and other available extractors.

Keywords
Keyphrase Extraction, Arabic Keyphrases, Information Retrieval, Classification Methods, Computational Linguistics.

1. INTRODUCTION
Automatic keyphrase extraction is the task to identify a small set of keyphrases, keywords, or key segments from a document [9]. Keyphrases are useful for many NLP applications such as indexing, text summarization, information retrieval, clustering, classification, web searches and others. With the frequently growing amount of electronic textual content available online, there is pressing need to have appropriate tools that can automatically extract keywords from text.

A recent survey of the state of the art in keyphrase extraction presented by Hasan and Ng [8] classified the extraction methods into two broad categories: supervised and unsupervised. The most keyphrase extraction systems which are proven to be successful have used supervised machine learning techniques [11]. In such techniques, a classifier is trained by using documents with known keywords (each keyword is classified either as a keyword or a non-keyword). The trained classifier is subsequently applied to predict new documents for which no keywords are assigned.

Some research works have been proposed and implemented for automatically extracting keyphrases for Arabic documents, but the proven efficiency was not satisfactory. Therefore, there is still an urgent need to make more efforts to enhance the performance of keyphrase extraction techniques for Arabic documents.

Extracting n-gram words results a huge number of candidate terms, many of these terms are not keyphrases. The problem we attack is how to limit these terms and keep the appropriate ones. The presented work is an extension to a previous one to extract keyphrases for Arabic documents. In this work, morpho-syntactic features of candidate terms are applied to a supervised machine learning technique to classify the candidates as keyphrases or not. The adopted features include suffix, prefix, POS of a term. To capture the well-structured terms, a new-added definite structure feature is introduced and tested.

This paper is organized as follows: In section 2, the related works and several techniques to keyphrases extraction are presented. Section 3 illustrates the details of proposed technique for keyphrase extraction. Section 4 shows the experimental work carried out to evaluate the performance of the presented technique. Finally, section 5 concludes the whole work.

2. RELATED WORK
Several keyphrase extraction techniques have been proposed and successfully implemented in different context. Keyphrase extraction can be seen as supervised learning from the examples. Turney introduces two supervised algorithms, a decision tree algorithm and a genetic algorithm [12], [13], [14]. The first algorithm is based on the C4.5 decision tree classifier [10], while the second is the GenEx (Genitor and Extractor) algorithm [12], [13], [14]. The phrase is represented as an n-gram (where n between 1 and 3 words) after stop words removal .In the learning task, twelve phrase features were used, such as the phrase length, phrase frequency, first occurrence of a phrase in a document, etc. The C4.5 algorithm was trained on nine of the twelve features, ignoring two features and using only one feature as a class prediction value.

The Keyphrase Extraction Algorithm KEA [6], [16], [17] uses the machine learning techniques and Naive Bayes algorithm to classify the candidate phrases as keyphrases or not. Candidate phrases are classified using four features: term frequency times inverse document frequency (TFxIDF), position in text, phrase length, and node degree of a candidate phrase (the number of phrases in the candidate set that is semantically related to this phrase). This is computed with the help of the thesaurus.

Both Turney's system and KEA uses surface level features. The work presented by Hulth [9] improves those supervised algorithms by using natural language processing features like part of speech tags. Terms are considered as keywords based on three statistical features in addition to a linguistic one: term frequency (TF), inverse document frequency (IDF), relative position of its first occurrence in adocument and part of speech.The results indicate that the use of linguistic features improves the automatic keyword extraction [7].

The Arabic Keyphrase Extractor AKE [3] uses a supervised machine learning technique to extract key phrases from Arabic text. The algorithm is based on combining statistical features with linguistic knowledge for better results. The linguistic

knowledge such as part-of-speech tags is used in building syntactic rules that filters the candidate keyphrases. Also, the abstract form of Arabic words is used instead of its stems. The system uses an annotated Arabic corpus for the specified domain. The learning model is built using Linear Discriminant Analysis (LDA). The system has better performance than KEA and Sakhr systems in terms of precision and recall obtained mainly from linguistic knowledge [3]. The previous work Lemma based Arabic keyphrase extractor LBAKE [5] –which the present work rely on- is an enhancement of the AKE [3]. The main modification is the replacement of the AKE' corpus-based module [3] with the Arabic lemmatizer module [4] to increase the coverage scope of the language analysis. The Arabic lemmatizer generates the lemma form and extracts the word features such as POS, prefix, and suffix.

3. THE PROPOSED TECHNIQUE

The proposed keyphrase extraction technique has two main phases. In the first phase, all the n-grams candidate keyphrase terms are extracted with their weights. This includes document analysis, and candidate Keyphrase extraction and scoring. The existing Arabic lemmatizer and LBAKE modules are adopted to perform these tasks. In the second phase, a supervised machine learning technique is applied to set of morpho-syntactic features of the candidate terms to classify them as keyphrases or not. The adopted features include suffix, prefix, POS of a term and its nearby terms. To achieve better results and capture the well-structured terms a new-added definite structure feature is introduced. The steps for the proposed technique are:

1. Document analysis: the document is tokenized into sentences and words. Each word is analyzed to extract its POS tags, category, and lemma form.
2. Candidate terms extraction and scoring: the sequence of n-gram terms with their corresponding weights is extracted.
3. Term classification: the extracted terms are reclassified as either keyphrases or not. The proposed technique is based on term's linguistic features and machine learning technique to extract the most expressive terms.

The details of such steps are illustrated in the following subsections:

3.1 Document Analysis

The existing Arabic lemmatizer [4] is utilized in performing all the necessary preparation linguistic analysis tasks. It accepts the input document, which is segmented into its constituent sentences and words. Each word is analyzed to generate its lemma form, and to extract its lexical features. The Arabic lemmatizer performs the following tasks:

1. Segments the document into its constituent sentences and words based on the Arabic phrases delimiter characters such as comma, semicolon, colon, hyphen, and dot.
2. Extracts POS tagging of the document words. Ambiguity is resolved using metadata about patterns, roots, and infixes' indications of Arabic words.
3. Transforms an inflected word form to its dictionary canonical lemma form. For nouns and adjectives, lemma form is the singular indefinite (masculine if possible) form, and for verbs, it is the perfective third person masculine singular form.
4. Extracts relevant morpho-syntactic features that support keyphrase extraction purposes.

The Arabic lemmatizer makes use of different Arabic language knowledge resources to generate accurate lemma form and its relevant features that support IR purposes. The lemmatizer algorithm is based on Arabic morpho-syntactic rules in addition to auxiliary dictionaries to enhance the lemmatizer performance. The Arabic lemmatizer identifies each word by its affixes (prefix, and suffix), pattern, root, lemma form, in addition to the linguistic features of the word such as category, gender, and count. The most beneficial outputs from the analysis process are concluded by the word category and lemma form. These features represent the basic linguistic knowledge required for the candidate phrases extraction process.

3.2 Candidate Terms Extraction and Scoring

The process of candidate keyphrases extraction and scoring is based on a previous work LBAKE. LBAKE is a supervised learning system for extracting keyphrases of single Arabic document.The extractor is supplied with linguistic knowledge as well as statistical information. The statistical features calculations are based on the lemma Arabic form of a word, which enhances all frequency-based statistical features, since it captures all words and phrases inflections.The linguistic and statistical features are used to learn the Linear Discriminant Analysis classifier to extract relevant keyphrases. All possible phrases of one, two, or three consecutive words that appear in a given document are generated as n-gram terms. These n-gram terms are accepted as candidate keyphrases if they follow syntactic rules that limit allowed POS sequences. The importance of a keyphrase (score) within a document is based on the following features:

1. Number of words in each phrase.
2. Frequency of the candidate phrase.
3. Frequency of the most frequent single word in a candidate phrase.
4. Location of the phrase sentence within the document.
5. Location of the candidate phrase within its sentence.
6. Relative phrase length to its containing sentence.
7. Assessment of the phrase sentence verb content.
8. Assessment as to whether the phrase sentence is in the form of a question.

Weights of these features were learned during building the classifier, and consequently were used in the keyphrases scoring calculations. The output of LBAKE is a set of scored candidate keyphrases normalized to their maximum, representing the input document.

The following syntactic filtering rules that limit allowed POS sequences are applied for extracting candidate phrases [3], [5]:

1. The candidate phrase can start only with some sort of nouns provided that not to be an adjective like general-noun, defined-noun, undefined noun, copulative noun and proper-noun.
2. The candidate phrase can end only with general-noun, place-noun, proper-noun, declined-noun, time-noun, augmented-noun, and adjective.

3. For three words phrase, the second word is allowed only to be a preposition, in addition to those cited in the above rule 2.

3.3 Term Classification

LBAKE algorithm adopted syntactic filtering rules that limits allowed POS sequences. These rules are used to prevent some undesirable morphological and syntactic patterns to be extracted as candidate keyphrases. We also adopt these rules in the extraction process. Such filtering rules improve the resulted keyphrases by prohibiting many of the terms that are not true keyphrases. However, after investigating different resulted output keyphrases it is noticed that there are a lot of undesirable extracted terms still appear and probably exist in the list of high score candidates. Indeed, two types of undesirable candidates can be observed: i) linguistically incomprehensive phrases and, ii) phrases that are already represented in other nearby phrases and there is no added value to their existence. Therefore, rather than rely on the previous rules only, we introduced a new technique to classify the previously extracted candidate terms as keyphrases or not. The incident classification mainly reflects the acceptance of the candidate terms linguistically, but does not necessarily address their importance in the specified document. The proposed technique classified the previously extracted candidates by applying their deep linguistic patterns to a machine learning technique. To capture the well-structured candidate terms, a new definite structure feature is introduced. The following subsections describe in details the proposed technique.

3.3.1 Defining the Features

A set of deep linguistic features are applied to machine learning algorithm to classify the previously extracted candidates and decide which terms are keyphrases and which are not.

The idea is to collect more information about each candidate term to classify it as keyphrase or not. Collected information for a term also includes features of its nearby terms (window ± 2) . The concept is that in n-gram method for the terms that are already exist in other terms, it is sufficient in many cases to select the more representative terms and ignore the others. Ten features are used to represent the feature vector of each candidate term. The attended features explore deep linguistic pattern of a term and its corresponding nearby terms. The features are: Length, Category, Word-Suffix, Word-Prefix, Previous, Pre-previous, Subsequent, Next-subsequent, Definite Structure, and Keyphrase Acceptance.

Length: number of words in the candidate term. As we are dealing with n-gram and n=3, the length is 1, 2, or 3.

Category: the POS of the candidate term tokens. In case of the term consists of several tokens, the POS is treated like a series. For example: (هندسة النظم - the systems engineering) has a category: indefinite Noun- definite noun.

Word-Prefix: prefix found within the candidate term tokens for example (ك، ب، و، ل، لل).

Word-Suffix: suffix found within the candidate term tokens, this includes the conjunction pronouns.

Previous: POS of the term before the current term, if the current term is already included in that term.

Pre-previous: POS of the term by the previous of the current term, if the current term is already included that term.

Subsequent: POS of the term after the current term, if the current term is already included in that term.

Next-subsequent: POS of the term after the following of the current term, if the current term is already included in that term.

Definite Structure: Morphological and syntactic features of the phrase are important indicators to determine the validity of the semantic senses of the language. In the proposed technique, a new expression called definite structure " تركيب معرفي" is introduced. We note that this is a new expression that does not exist in the Arabic language resources. Definite structure feature conveys a set of linguistic rules mapped only in one feature. The definite structure term is the term that can only include proper nouns, definite nouns, compound structure, or propositional phrase.

We note here to some clarifications and points that were considered in the computations:

- Compound structure " تركيب اضافى " is a famous Arabic linguistic term, it is composed of two words (tokens) with the POS sequence indefinite noun - definite noun.
- The propositional phrase that we take into consideration is composed of two words (tokens) with POS sequence preposition-noun.
- The compound structure phrase is treated during the computation as a definite noun.
- A definite noun is concerned only with the case of noun with a prefix “AL” "معرف بال".

An example of extracted candidate phrases that have definite structure is as follows:

البيانات، قواعد البيانات، التعليم في مصر، إدارة مشاريع البرمجيات

Note that the candidate term "إدارة مشاريع البرمجيات " includes the compound structure " مشاريع البرمجيات " which is treated as definite noun, in turn when added to the word "إدارة " composed other compound structure which has definite structure.

The notion of definite structure arises after investigating different human written keyphrases for Arabic documents. Definite structure represented as a binary feature in the sense that according to the phrase morpho-syntactic structure it is classified as either positive or negative definite structure (1 or 0). The feature is computed by exploiting the extracted output produced by the Arabic lemmatizer. To investigate the feature importance, the learning experiments are applied excluding and including this feature in the term feature vector.

Keyphrase Acceptance: This feature is used during the training phase. It is manually assigned as a positive or negative class. It has a value of one if the candidate phrase is accepted as a keyphrase; otherwise it will be zero.

3.3.2 Learning Experiments

In the previous stage, the given document is converted to array of candidate terms; each candidate is represented by a vector of features. The input to the learning algorithm consists of examples; each example is represented by a feature vector for a candidate term and is manually classified as a positive or negative class. Then, based on the input examples, the system builds the classifier which is used to classify a new set of candidates either keyphrases or non-keyphrases. The output from the machine learning algorithm is in the binary form (a candidate term is either a keyphrase or not).

3.3.3 Training data

A dataset of ten documents is used to train the classifier. The training data are collected from different sources and domains with focus on the computer area. After preprocessing and extracting candidate terms, each candidate term is manually classified either keyphrase or not. The manual classification of the candidate terms are based mainly on two factors :i) semantic and linguistic acceptance of a term, and ii) term meaning value added of in case of the repetition in other nearby terms. The training sample produces 4153 candidate terms. The manual assigned terms are mapped in the Keyphrase acceptance feature and classified as 1617 positive and 2536 negative keyphrases. The dataset was spitted 70.0% as training data set, and the remainder as testing data set.

3.3.4 Building the classifier

In this work, the WEKA platform [1], [2], [15] is used to classify the candidate terms as keyphrases or not. We have examined three different classification models: J48, Decision Table, and Naïve Bayes. To test the importance of the new definite structure feature, the data set is applied including and excluding this feature. Tables 1& 2 illustrate the results of the three classifying techniques with the exclusion and inclusion of the definite structure feature. The results show an improvement obtained by including the definite structure feature in terms of correctly classified instances, precision, recall, and F-measure. The precision, recall, and F-measure results of the three classifiers are close; the best results (Correctly Classified Instances = 88.45 %, F-Measure = 0.89) are obtained by the inclusion of the definite structure feature and applying Decision Table classifier.

Table 1: Results excluding the definite structure feature

	Correctly Classified Instances	Precision	Recall	F-Measure
J48	**85.63 %**	**0.87**	**0.86**	**0.86**
Decision Table	84.77 %	0.85	0.85	0.85
Naïve Bayes	83.30%	0.86	0.83	0.84

Table 2: Results including all features

	Correctly Classified Instances	Precision	Recall	F-Measure
J48	87.96 %	0.90	0.88	0.88
Decision Table	**88.45 %**	**0.90**	**0.89**	**0.89**
Naïve Bayes	86.24**%**	0.88	0.86	0.87

4. PERFORMANCE EVALUATION

To measure the performance of the proposed technique and investigate the effect of the new added phase to the original extractor LBAKE, two experiments were applied. The first experiment was used to test the validity of the definite structure feature. In the second experiment, the extracted keyphrases by the proposed technique were compared to those extracted by the LBAKE version to test the proposed technique. The following subsections describe more details of the two experiments.

4.1 Experiment 1

In this experiment, the same training data set (described in section 3.3.3) is applied to test the validity of the new definite structure feature. The experiment starts by extracting the candidate terms and then calculating the definite structure feature. As mentioned previously, the definite structure feature expresses the validity of the candidate term as keyphrase or not. Then, the number of correctly classified instances is counted as the number of matched results occurs between the definite structure feature and the manual term classification. The average accuracy obtained was 83.56%, measured as the total number of the correctly classified instances among all instances. The gained accuracy is considered a satisfactory result compared to the best (88.45 %) and the lower results (83.30%) achieved by the machine learning technique reported in the previous section. This verifies that the definite structure feature can be used to stand alone as an inexpensive additional filtering stage to capture the well-structured terms for traditional statistical extraction techniques.

4.2 Experiment 2

Two data sets are used in this experiment to compare the output extracted keyphrases by the proposed technique against the LBAKE version. The first is the same dataset described in [3]. The second is a set of twenty documents collected from different web sources in scientific domains. The documents are manually keyphrase-assigned by scientific specialists. The average length of documents is 426.5 words. To measure the performance of the two systems, Precision, Recall, and F-measure are calculated. Table 3 shows sample of the extracted keyphrases for the two systems. The results in table 3 show the improvement in both of the number of author matched keyphrases and the keyphrase linguistic content as well. It is noticed the improvement of the extracted keyphrases, they convey well-structured informative content; even they are not author-matched keyphrases. Table 4 shows the precision, recall and F-measure for the two systems. It is clear from the results that the proposed technique has on the average better performance (F-measure =0.62) than the original LBAKE version (F-measure =0.56). The additional benefit we get is capturing the most representative and well-structured phrases and limiting the others. The exclusion of non-keyphrases from the list of candidates allows a room for other good kephrases to be presented in the output. The experimental results verified that the use of deep linguistic features of the candidate terms improves the results of the extracted keyphrases. It is mentioned that the experimental results reported in [5] show that LBAKE has the superiority over other two Arabic keyphrase extraction systems: KP-Miner (web link http://www.claes.sci.eg/coe_wm/kpminer), and Sakhr Keyword Extractor (web link http://www.sakhr.com/ Technology/ Keyword/ Default.aspx? sec=Technology &item= KeywordS).

Table 3: Sample of the extracted keyphrases of the two systems, with those matching author keyphrases are underlined and bolded.

The Proposed Technique	LBAKE
الاختبار-**الاختبار الالي** -**اختبارالصندوق الأسود**-**اختبار الصندوق الأبيض** -أدوات اختبار البرمجيات-**مراقبة الجودة** -**اختبار البرمجيات**- اختبار محتوى المنتج- أساليب الاختبار	الاختبار-**الاختبار الالي** -اختبار الصندوق-**اختبار الصندوق الأسود** -**اختبار الصندوق الأبيض**-الجودة- متخصص في اختبار -أدوات اختبار - استخدام أدوات اختبار -**مراقبة الجودة**
الطاقة- الطاقة الكيميائية- **مصادر الطاقة**- الطاقة الكيميائية المختزنة- **حفظ موارد الطاقة**- تحويل طاقة الشمس- طاقة الشمس- نتيجة لتحولات الطاقة- **تحولات الطاقة**- **موارد الطاقة**	**الطاقة**- الطاقة الكيميائية- **مصادر الطاقة**- الطاقة الكيميائية المختزنة- كلمة طاقة- **حفظ موارد الطاقة**- طاقة كيميائية- تحويل طاقة الشمس- طاقة الشمس- **تحويل الطاقة**
الحيوانات- **مملكة الحيوان** - النباتات- **مملكة الحيوان**- النباتات -إختلاف الحيوانات- المخلوقات - **حركة الحيوانات**- **الحواس**	**الحيوانات**- **مملكة الحيوان** - النباتات- أنواع- طريق- الحواس بإختلاف الحيوانات- مملكة.

Table 4: Sample of average results for the two systems

	Precision	Recall	F-Measure
LBAKE	0.48	0.69	0.56
The Proposed Technique	0.54	0.76	0.62

5. CONCLUSION

In this paper, I have shown that keyphrases extraction for Arabic documents can be improved by using deep linguistic knowledge. This work is based on the existing LBAKE Arabic keyphrase extractor. The proposed technique classified the candidate terms by applying their deep linguistic patterns to a machine learning technique. Ten features are used to represent the feature vector of each candidate term. To capture the well-structured candidate terms, a new-added definite structure feature was introduced and tested. Two experiments were carried out. The first experiment was used to test the validity of the definite structure feature. The results of this experiment verified that the definite structure feature can be used to stand alone as an inexpensive additional filtering stage to capture the well-structured terms for traditional statistical extraction techniques. The second experiment was used to compare the output keyphrases extracted by the proposed technique against the LBAKE version. The experimental results showed that the proposed technique has a significantly better performance than that of LBAKE. The F-measures reported were 0.56 and 0.62 for LBAKE and the proposed technique respectively. The additional benefit we get is capturing the most representative and well-structured phrases. The experimental results verified that the use of deep linguistic features of the candidate terms helps improve the accuracy of extracted keyphrases.

6. REFERENCES

[1] Bouckaert R, Eibe F. 2010. WEKA Manual for Version 3.7.12.

[2] CBA. Data mining tool Downloading page at : http://www.comp.nus.edu.sg/~dm/p_download.html.

[3] El-Shishtawy, T. , Al-sammak, A. 2009. Arabic Keyphrase Extraction using Linguistic knowledge and Machine Learning Techniques. Proceedings of the Second International Conference on Arabic Language Resources and Tools. The MEDAR Consortium, Cairo, Egypt.

[4] El-Shishtawy, T., El-Ghannam, F. An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes. International Journal of Computer Science Issues, 2012, Volume 9.

[5] El-Shishtawy, T. , El-Ghannam, F. 2012. Keyphrase Based Arabic Summarizer (KPAS). 8th International Conference on Informatics and Systems INFOS, Egypt.

[6] Frank, E., Paynter,W., Witten, H., Gutwin, C., and Nevill-Manning, G. 1999. Domain-specific keyphrase extraction. Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99), pp. 668-673. California: Morgan Kaufmann.

[7] G¨onen, E. Automated text summarization and keyphrase extraction. 2006. Master thesis, Bilkent University.

[8] Hasan, S., NG, V.: Automatic keyphrase extraction. 2014. A survey of the state of the art. Proceedings of the Association for Computational Linguistics (ACL), Baltimore, Maryland: Association for Computational Linguistics.

[9] Hulth, A. 2003. Improved automatic keyword extraction given more linguistic knowledge. Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Sapporo, Japan.

[10] Quinlan, R. C4.5: Programs for Machine Learning. Morgan Kaufmann, Los Altos, (1993).

[11] Sarkar, K.: A hybrid approach to extract keyphrases from medical documents. International Journal of Computer Applications . 2013 , (0975 – 8887) Volume 63– No.18.

[12] Turney, D. 2000. Learning algorithms for keyphrase extraction. Information Retrieval, 2, pp.303-336.

[13] Turney, D. 1999. Learning to Extract Keyphrases from Text. National Research Council, Institute for Information Technology, Technical Report ERB-1057.

[14] Turney, D. 1997. Extraction of Keyphrases from Text: Evaluation of Four Algorithms. National Research Council, Institute for Information Technology, Technical Report ERB-1051.

[15] Witten, I., frank, E. 2000. Data Mining: practical machine learning tools and techniques with Java implementations, Morgan Kaufmann‖, San Francisco.

[16] Witten, I., Paynter, W., Frank E., Gutwin C. and Nevill-Manning. 1999. KEA: Practical Automatic keyphrase extraction. Proceedings of Digital Libraries 99 (DL'99), pp. 254-256. ACM Press.

[17] Witten, I., Paynter, W., Frank, E.,Gutwin, C., and Nevill-Manning, G. 2000. KEA: Practical Automatic Keyphrase Extraction. Working Paper 00/5, Department of Computer Science. The University of Waikato.